

いずれかのクラスタサーバが起動したとき、マスタノード（サーバ）は最初にクラスタを形成する必要があります。Cluster Volume Manager を実行しているクラスタノードは、起動プロセスで Cluster Volume Manager のボリューム再構成デーモンを起動する必要があります。このデーモンは、クラスタ共有ディスクグループを検出すると、それらのディスクグループを自動的にインポートし、それらのボリュームをノードからアクセスできるようにします。クラスタ共有ディスクグループを最初にインポートしたノードは、そのグループのマスタノードになります。追加のサーバがクラスタに参加した場合、それらのサーバはクラスタ共有ディスクグループをインポートし、そのディスクグループのクライアントユーザーになります。Cluster Volume Manager を備えたクラスタ内のすべてのサーバは、起動時にすべてのクラスタ共有ディスクグループをインポートします。

クラスタ共有ディスクグループのマスタノードは、ディスクグループのすべてのメタデータ変更を行います。メタデータ変更には、たとえば、ディスクグループ内部のストレージ容量を使用したボリュームの作成、ボリュームのサイズ変更、ミラー化ボリュームからのミラーコピーの削除などが含まれます。ディスクグループのメタデータを変更した場合、マスタは必ず分散トランザクションの手法を使用してすべてのノードの同期をとり、変更がクラスタ全体で同時に有効となるようにします。

Before one of the various cluster servers start up, the master node (server) needs to form a cluster first. The cluster node that is running the Cluster Volume Manager needs to start up the volume reconfiguration daemon as an active process in the Volume containing the Cluster Volume Manager. When this daemon detects the cluster sharing disc group, it imports them and enable the access of these volumes from the node automatically. The node that first imports the cluster sharing disc group becomes the master node for the group. If an additional server joins the cluster, the cluster server imports the cluster sharing disc group which becomes a client user of the existing disc group. All the servers that have Cluster Volume Manager in the cluster imports all cluster sharing disc groups when they start up.

The master node for the cluster sharing disc group executes all metadata changes in the disc group. The following are examples of metadata changes; volume creation that uses the disc group's storage capacity, volume size change, mirror copy deletion from a mirrored volume, etc. In case where the metadata changes in the disc group, the disk group master always initiates the synchronization of all the nodes using a decentralized transaction method and makes the change available to the whole cluster at the same time.

**VERITAS Cluster Volume Manager
VERITAS Cluster File System**

クラスタ環境のための新しい
VERITAS のボリューム管理
およびファイルシステム
テクノロジ

目次

I. VERITAS Cluster Volume Manager.....	3
VERITAS Volume Manager.....	3
VERITAS Cluster Volume Manager.....	5
Cluster Volume Manager のアーキテクチャの概念.....	5
クラスタ共有ボリュームサームサーバとクライアントノード.....	6
Cluster Volume Manager の機能の概要.....	7
クラスタボリュームの意味.....	7
Cluster Volume Manager とシステムの障害.....	8
II. VERITAS Cluster File System.....	9
クラスタのファイルシステム.....	9
Cluster File System のプロパティ.....	10
VERITAS Cluster File System の利点.....	11
Cluster File System とアプリケーション.....	12
VERITAS Cluster File System のアーキテクチャ.....	12
サーバクライアントのファイルシステム設計.....	12
Cluster File System の耐障害性について.....	13
Cluster File System と VERITAS Global Lock Manager.....	14
Cluster File System と VERITAS Cluster Server のプロトコル.....	14



図 1: VERITAS Volume Manager のボリュームアーキテクチャ.....	3
図 2: VERITAS Cluster Server と Volume Manager ボリュームによる構成.....	4
図 3: プライベートディスクグループとクラスタ共有可能ディスクグループ.....	6
図 4: Cluster Volume Manager 上に階層化された VERITAS Cluster File System.....	9
図 5: VERITAS Cluster File System のサーバクライアントアーキテクチャ.....	13
図 6: VERITAS Cluster File System に統合されているコンポーネント.....	15

VERITAS Cluster Volume Manager

VERITAS Volume Manager は実用上きわめて便利であることが実証されていますが、その単一制御ボイントモデルは、共有ディスクスタをサポートしていません。このため、ペリタスソフトウェアでは VERITAS Volume Manager をベースにした Cluster Volume Manager を開発し、VERITAS Cluster Server またはその他 (Oracle Parallel Server など) の共有データクラスタ用に堅固な共有ボリュームデクノロジーを提供しています。

Cluster Volume Manager は、Volume Manager をベースとした製品で、次の機能が追加されています。(SANPoint Foundation Suite など、いくつかの製品に組み込まれて提供されます。)

- 複数のサーバーから各ボリュームへの同時アクセス
- クラスタ全体の論理デバイスのネーミング
- すべてのサーバーで整合性のある、ボリューム状態の論理ビュー
- クラスタ内の任意のサーバーからのボリューム管理
- サーバに障害が起きた後、アクセス可能で残ったボリュームへの生き残ったサーバーからのアクセス
- ボリュームフェイルオーバーのないアプリケーションジョブメールオーバー

Volume Manager と同様、Cluster Volume Manager は、物理ディスクと、ハードウェア RAID アレイサブシステムによりエクスポートされた仮想ディスクの両方を管理できます。

Cluster Volume Manager のアーキテクチャの概念

Volume Manager と同様、Cluster Volume Manager はディスクをディスクグループに編成します。それぞれのボリュームは、一つのディスクグループに属するディスクから割り当てられます。Cluster Volume Manager のディスクグループであるボリュームは、プライベート、あるいはクラスタ共有の属性を持ちます。ボリュームが物理的にクラスタ全体に接続されている、プライベートディスクグループに属するディスクは 1 台のサーバーからアクセスできます。図 3 の場合、A、B、C、D の各ディスクグループは、それぞれが単一のサーバーに接続されているので、必然的にプライベートルートディスクグループとなります。各サーバーは、サーバーごとに一つのプライベートルートディスクグループを持つ必要があり、そのことは、そのグループに属するディスクが物理的に複数のサーバーに接続されているかどうかとは関係ありません。

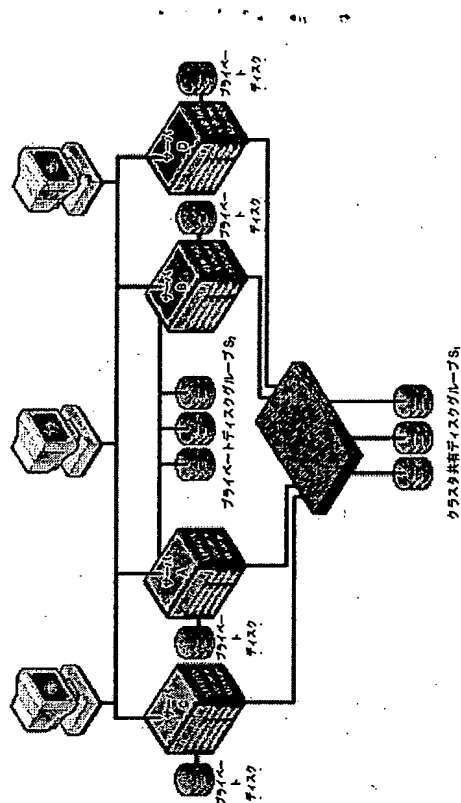


図 3: プライベートルートディスクグループとクラスタ共有可能ディスクグループ

クラスタ共有ディスクグループは、複数のサーバーから同時にアクセスできます。これらのディスクグループは、物理的にクラスタ内のすべてのサーバーに接続されている必要があります。図 3 の場合、ディスクグループ S1 はクラスタ共有ディスクグループにすることができ、ディスクグループ S2 はサーバー A とサーバー B だけがアクセスできるので、クラスタ共有にできません。ディスクグループ S2 は、サーバー A とサーバー B のどちらかが所有するプライベートルートディスクグループとして指定できます。

Cluster Volume Manager は、単純ボリューム (単一ディスク)、スパンボリューム (連続)、ミラー化ボリューム、ストライプ化したミラー化ボリューム (単一ディスク)、およびミラー化したストライプ化ボリュームをサポートします。今後、パリティ RAID ボリュームのサポートを予定しています。(時期未定)

クラスタ共有ボリュームサーバーとクライアントノード

VERITAS Volume Manager の場合と同様に、システム管理者は vvdg ユーティリティを使用してディスクグループをプライベートルートとクラスタ共有のどちらかに指定します。プライベートルートディスクグループは、機能的には単一ホストシステム上の VERITAS Volume Manager ディスクグループと同じものです。

いずれかのクラスタサーバーが起動したとき、マスタノード (サーバー) は最初にクラスタを形成する必要があります。Cluster Volume Manager を実行しているクラスタノードは、起動プロセスで Cluster Volume Manager のボリューム再構成デーモンを起動する必要があります。このデーモンは、クラスタ共有ディスクグループを検出すると、それらのディスクグループを自動的にインポートし、それらのボリュームをノードからアクセスできるようにします。クラスタ共有ディスクグループを最初にインポートしたノードは、そのグループのマスタノードになります。追加のサーバーがクラスタに参加した場合、それらのサーバーはクラスタ共有ディスクグループをインポートし、そのディスクグループのクライアントユーザーになります。Cluster Volume Manager を備えたクラスタ内のすべてのサーバーは、起動時にすべてのクラスタ共有ディスク

グループをインポートします。

クラスター共有ディスクグループのマススタノードは、ディスクグループのすべてのメタデータ変更を行います。メタデータ変更には、たとえば、ディスクグループ内部のストレージ容量を使用したボリュームの作成、ボリュームのサイズ変更、ミラー化ボリュームからのミラーコピーの削除などが含まれます。ディスクグループのメタデータを更新した場合、マススタは必ず分散トランザクションの手法を使用してすべてのノードの同期をとり、変更がクラスター全体で同時に有効となるようにします。

Cluster Volume Manager の機能概要

Cluster Volume Manager のリリース 3.2 は、最大 16 台のサーバー間のボリュームの共有（つまり 16 ノードのクラスター）をサポートしています。Cluster Volume Manager は、複数のバス（I/O パス）上にあるディスクアレイにアクセスできる RAID コントローラや、すべてのクラスターノードのアクセスパスを同時に切り替えることができる RAID コントローラを認識できます。「ローリング」ソフトウェアアップグレードをサポートされるので、Cluster Volume Manager のインストレーションは、一度に一つのクラスターノードをサポートから外すだけで更新できます。

クラスターボリュームの意味

Cluster Volume Manager は、名前が示すとおり、クラスターボリュームマネージャです。Cluster Volume Manager はボリュームを、ディスクと同じように、クラスターファイルシステムやその他のシステムアプリケーション、たとえば分散データベースマネージャなどで使用するためにエクスポートします。Cluster Volume Manager はメタデータの更新の同期をとり、クラスター共有ディスクグループのボリュームについて、クラスター全体で整合性のあるビューを提供します。しかし、ボリューム自体は、アプリケーションからの読取りコマンドや書き込みコマンドに対してディスクのように動作します。

ボリュームは、ディスクのように、ホストの読取りコマンドと書き込みコマンドを無差別に実行します。サーバーは、ディスクに物理的にアクセスできる場合、ディスクに対して、ボリュームの場合と同じように、有効なすべての読取りコマンドまたは書き込みコマンドを実行できます。この動作は、一つのディスクまたはボリュームが、そのボリュームへのすべてのアクセスを調整する一つのファイルシステムまたはデータベースシステムによってマウントされる単一サーバーシステムでは適切な動作です。しかし、クラスターでは、複数のノードが同時に一つのディスクの読取りや書き込みを行う場合があります。ノード間の調整を行わないと、データの更新が同時に相対互に上書きされるおそれがあり、同じブロック範囲への別のノードが書き込みを行っているときに読取りを実行すると、整合しない「移動中の」データイメージが返ってくるおそれがあります。

Cluster Volume Manager は、上記のように、このディスクとして読込みと書き込みを実行します。しかし、Cluster Volume Manager は、異なるクラスターノードのユーザーによるデータの読込みが相互に干渉しないことを保証するわけではありません。そのような保証は、ボリュームを使用するファイルシステムまたはデータベースマネージャが行う必要があります。Cluster Volume Manager は、クラスターに対応したデータマネージャ、たとえば、VERITAS Cluster File System などと併用するように設計されています。このデータマネージャは、ファイルやディスクブロックなど、クラスター全体での共有データオブジェクトへのアクセスを調整する分散ロックマネージャを備えています。Cluster File System は、VERITAS Global Lock Manager を使用してデータアクセスを調整します。

Cluster Volume Manager とシステムの障害

Cluster Volume Manager は、VERITAS Volume Manager テクノロジーを使用して二つ以上のディスクにデータのミラー化を行い、ディスクの耐障害性の向上させます。ボリュームをミラー化すると、Volume Manager はアプリケーションの書き込みを、そのボリュームに対応するそれぞれのディスクのブロックに対して行います。

サーバーは、ミラー化が進行中のボリュームへの書き込みに失敗することもあります。それにより、更新されたコピーとそうでないコピーが生じ、ボリュームの内容が整合しなくなることが考えられます。

制御ポイント（ボリュームマネージャを実行しているサーバー）がクラッシュした後に耐障害性の高いボリュームの内容の整合性を保証することは、ボリューム管理テクノロジでは既知の問題であり、解決方法もいくつか知られています。たとえば、ミラー化ボリュームをクラッシュ後に整合させるには、一つのディスクをマススタとして指定し、そのすべての内容を、一緒にミラー化された各ディスクにコピーします。しかし、このような解決方法は時間がかかり、データとアプリケーション双方の可用性を低下させます。

ペリタスソフトウェアの製品は、システムがクラッシュした後、ボリュームとファイルシステムの整合性を迅速に復元できるように、ミラー化ボリュームの場合、DRL（ダーティリージョンログ）は、ボリュームのどのブロック範囲（リージョン）が最近更新されたかを示し、その結果、システムがクラッシュの後、どのブロック範囲が整合していない可能性があるかを示します。DRL のエントリは、ミラー化ボリュームのディスクが更新される前に書き込まれます。あるボリュームのすべてのミラーが更新された場合、そのリージョンは、もはや危険でないものとしてマークされます。オーバーヘッドを最小限に抑えるため、このログへの書き込みは、保留中の別の更新が必要とするまで行われません。

Cluster Volume Manager は、ノードのクラッシュからリカバリすると DRL を読取り、危険のフラグが付いているミラー化ボリュームリージョン内のデータを再コピーします。ある時点で危険となるボリュームのリージョンは小さな断片にすぎないと考えられるので、リカバリ時間は、非常に大きなボリュームでも、追加的なコピーの場合（断片または数時間）よりずっと短く、数秒で済みます。

II. VERITAS Cluster File System

クラスタのファイルシステム

VERITAS Cluster Volume Manager は、クラスタ全体にわたり、堅固な論理ボリュームをデータマネージャと Raw デバイスアプリケーションからアクセスできるようにします。VERITAS Cluster File System を使用すると、一つのファイルシステムを複数のクラスタサーバで同時にマウントして使用でき、そのファイルシステムを使用するすべてのアプリケーションが同じサーバ上で実行されているのと同じこととなります。図 4 は、クラスタ内の Cluster File System を示しています。

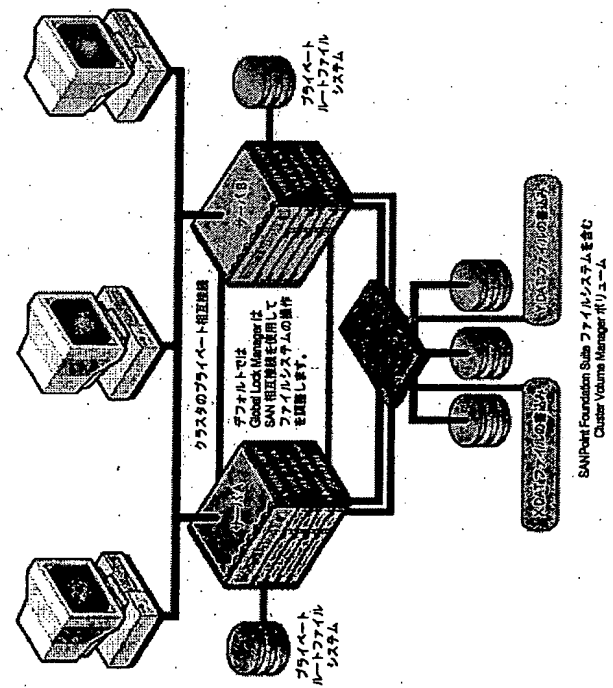


図 4: Cluster Volume Manager 上に構築された VERITAS Cluster File System

Cluster File System のプロパティ

VERITAS Cluster File System は、単一ホストの VERITAS File System をベースとしています。VERITAS File System は、その成熟度と豊富な機能セットにより、エンタープライズ UNIX 環境に特に推奨するファイルシステムです。VERITAS File System がエンタープライズ向けアプリケーションとして特に有効である理由は次のとおりです。

- テラバイトのサイズまでのファイルを簡単にマップできるエクステンシブルスペース管理。
- ほとんどのシステムクラッシュからの迅速なリカバリ。これには、最近のファイルシステムメタデータの更新を追跡する自己クリア方式のインテントログが使用されます。
- ファイルシステムを使用しながら拡張とデフラグメントができるオンライン管理機能。
- Quick I/O 機能。これにより、対応するデータスペースマネージャでファイルの Raw パーティションとして扱うことによりカーネルロックを迂回できます。また、Quick I/O の拡張機能である Cashed Quick I/O では 32 ビットアプリケーションで 4 ギガバイトを超えるシステムキャッシュを利用できます。(この機能は、Database Edition にて提供されます。)

VERITAS Cluster File System を使用すると、これらすべての機能をサーバのクラスタから使用できます。また、Cluster File System は、クラスタ環境を利用して次の機能も提供します。

- クラスタ全体にわたるファイルシステムの状態のスナップショットの作成。これにより、ファイルシステムの一貫したオンデマンドイメージを必要とするオペレーション (たとえば、バックアップ、用またはテスト用のミラー化ボリュームからコピーを削除するなど) をクラスタ環境で行うことができます。
- クラスタ全体とローカルの両方におけるファイルシステムのマウント。これにより、管理者はクラスタノード間でデータを共有したり、アプリケーション要件による制約を離れてデータを共有したりできます。
- Cluster File System 自体の「ローリング」更新。これにより、Cluster File System をノードごとに更新でき、アップグレードプロセス全体を通して、クラスタを一体のものととして操作できます。

Cluster File System ファイルシステムを使用しているサーバに障害が起きた場合、そのサーバのアプリケーションが、残存するサーバへフェイルオーバーする可能性があります。多くの場合、ファイルシステムの再起動は必要ありません。なぜなら、ファイルシステムは依然として稼働しているからです。このため、システムクラッシュの後にアプリケーションを再起動したとき、通常は時間消費の主要な原因となるアプリケーションデータのリカバリ作業を排除できます。

VERITAS Cluster File System の利点

SANPoint Foundation Suite と SANPoint Foundation Suite HA では、次に述べるように、ハードウェアの制限から生じる大規模システムでの管理作業の多くが簡素化されるか不要になります。

- Cluster Volume Manager はテラバイトのボリュームの作成と管理ができるので、ファイルシステムをディスク容量の限度内に収めるようにパーティション分割する必要はほとんどありません。
- Cluster File System は 1 テラバイトまでの容量のファイルシステムをサポートできるので、パーティション分割が必要となるのは、非常に大量のデータの転送を使用し、ファイルシステムのアドレス指定の制限のために分割が必要となる場合だけです。
- クラスター内のすべてのサーバが Cluster File System のクラスター共有可能ファイルシステムにアクセスできるので、参照データまたはアプリケーションのイメージとライブラリの整合性を複数のサーバにわたって維持することは自動で行われます。すべてのクラスターノードが、同じ参照データおよびイメージから作業を行うことができます。それだけでなく、非共有データクラスターでアプリケーションと参照データの複数の同一コピーに必要となるストレージの容量は、すべてのサーバが同じデータとイメージから作業を行うときは不要になります。
- すべてのサーバがすべてのファイルにアクセスできるので、アプリケーションを各サーバに割り当て、負荷を均等化したり、その他のオペレーション要件を満たすことができます。同様に、フェイルオーバーも、データのアクセス可能性により制約を受けないため、柔軟に行うことができます。
- すべてのクラスターノードが各 Cluster File System のマスタになり得るので、ファイルシステムのマスタ機能をクラスターノード間で分散することにより、フェイルオーバー時に占めるファイルシステムのリカバリ部分を、 n 個のノードからなるクラスターでは n 分の 1 に減少させることができます。
- エンタープライズ RAID サブシステムを投資に見合うものにし、より効果的に使用できます。なぜなら、それらのサブシステムのストレージ容量をすべてのサーバがマウントでき、変化するビジネスの要求に合わせて、ハードウェアの再構成ではなく管理オペレーションにより再割り当てができるからです。
- ファイルシステムを共有することにより、ストライプ化の幅が広がり、より大きなボリュームが使用可能となり、アプリケーションの I/O ロードバランシングが向上します。それぞれのサーバの I/O 負荷がより多くの I/O リソースに分散されるだけでなく、Cluster File System 共有ファイルシステムを使用すると、すべてのサーバの負荷がすべての I/O リソース間で均等化されます。
- ボリュームとファイルシステムの数が少なくなるため、管理者、ユーザー、アプリケーションの開発者と管理者は個々のオブジェクトを見つけるのが容易になります。またバックアップについても同様です。
- サーバの追加によるクラスターの拡張が容易になります。なぜなら、それぞれの新規サーバのデータストレージ構成は、ほとんどの場合、事前定義されているからです。新規サーバは単に既存のクラスター全体のボリュームとファイルシステムの構成を適用することにより可能になります。
- Cluster File System の単一システムイメージ管理モデルは、すべてのファイルシステム管理オペレーション（サイズ変更は除く）をそれらのオペレーションが呼び出された場所に依存しないものにする（ただし、痕跡のメタデータオペレーションは現行の Cluster File System マスタノードにより実行される）ので、管理を簡素化します。
- 管理者は、使用可能なサーバ容量と処理するデータがありながら、その二つを結び付ける方法がな

Cluster File System の単一システムイメージ管理モデルは、すべてのファイルシステム管理オペレーション（サイズ変更は除く）をそれらのオペレーションが呼び出された場所に依存しないものにする（ただし、痕跡のメタデータオペレーションは現行の Cluster File System マスタノードにより実行される）ので、管理を簡素化します。

という状況を回避できます。Cluster File System クラスター共有可能ファイルシステムを使用すると、すべてのサーバが、実質的に単一サーバファイルシステムと同じように、すべてのデータにアクセスできます。

- SANPoint Foundation Suite および SANPoint Foundation Suite HA の新しいオペレーション機能である VERITAS FlashSnap™ により、データの分析、バックアップ、テストおよびレポート機能などのオペレーションを運用環境に影響を及ぼすことなく実行できます。

Cluster File System とアプリケーション

多くのアプリケーションが Cluster File System の恩恵を受けることができます。従来の「クラスター非対応」アプリケーションも、クラスター内のすべての場所で実行でき、すべての場所のデータにアクセスできます。マルチアプリケーションクラスターでは、Cluster File System ファイルシステムが比較的大きなボリューム上に構築されていれば、ロードバランシングの向上により全体的な I/O パフォーマンスが向上します。これらの利点は、Cluster File System をインストールすれば自動的に得られます。チューニングやその他の管理操作は必要ありません。

多くのアプリケーションはパーティション分割が可能です。つまり、それらは複数の同時実行スレッドから構成され、それらのスレッドは、それぞれのデータアクセスを調整する方法があれば、異なるサーバ上でも実行できます。Cluster File System は、その調整を行います。そのようなアプリケーションは「クラスター対応」にすることができ、インスタンスを複製させてクライアントとデータアクセスの負荷を均等化し、それにより、単一サーバの容量を超えた拡張を行うことができます。そのようなアプリケーションでは、Cluster File System は I/O の負荷を均等化するだけでなく、共有データアクセスを提供することにより、クラスターノード間でアプリケーションレベルのロードバランシングを可能にします。

VERITAS Cluster File System のアーキテクチャ

サーバクライアントのファイルシステム設計

VERITAS Cluster File System は、マスタクライアントのアーキテクチャを使用し、共有ボリューム上のファイルシステムメタデータを管理します。それぞれの Cluster File System ファイルシステムを最初にマウントしたサーバが、そのファイルシステムのマスタになります。クラスター内の他のノードは、すべてクライアントになります。アプリケーションは、図 5 に示すように、そのアプリケーションを実行しているサーバから直接、ファイル内のユーザーデータにアクセスします。しかし、Cluster File System ファイルシステムメタデータを更新できるのは、ファイルシステムのマスタノード（そのファイルシステムを最初にマウントしたノード）だけです。Cluster File System マスタノードはすべてのメタデータの更新に責任を負い、ファイルシステムのメタデータ更新インテグリティの保守にも責任を負います。クライアントサーバは、ファイルシステムメタデータを更新する必要がある場合（たとえば、新しいファイルを割り当てたり古いファイルを削除する必要がある場合）、マスタにその要求を伝え、マスタは実際の更新を行い、要求元のサーバに返答します。この設計により、ファイルシステムメタデータの整合性と、システムクラッシュからのリカバリに使用されるインテグリティログの整合性が保証されます。

Cluster File System の単一システムイメージ管理モデルは、すべてのファイルシステム管理オペレーション（サイズ変更は除く）をそれらのオペレーションが呼び出された場所に依存しないものにする（ただし、痕跡のメタデータオペレーションは現行の Cluster File System マスタノードにより実行される）ので、管理を簡素化します。

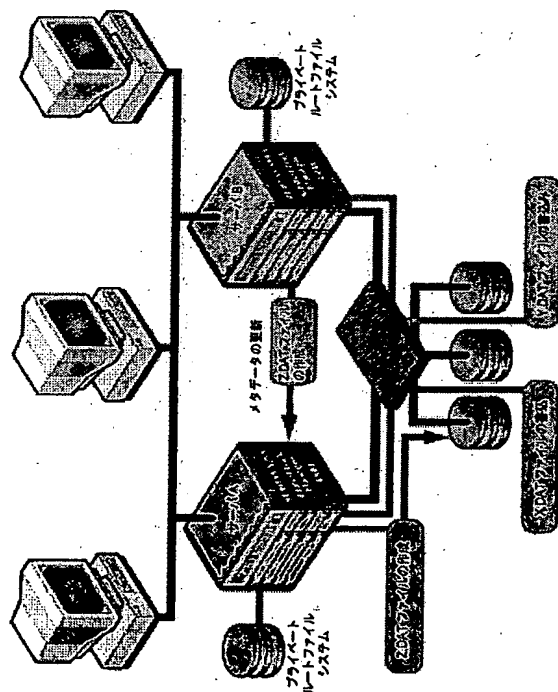
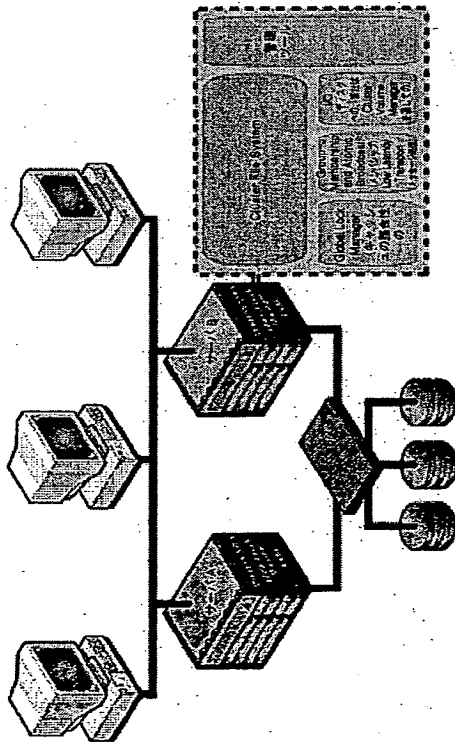


図 5 : VERITAS Cluster File System のサーボクライアントアーキテクチャ

Cluster File System の耐障害性について

Cluster File System マスタサーバに障害が起きた場合、残存するクラスタノードは新しいマスタを選出します。新しいマスタはファイルシステムのインテントログを破取、障害が起きたときに処理中だったメタデータへの更新があれば、それを完了します。アプリケーション I/O は、リカバリの間、停止され、通常ではリカバリまでに数秒かかります。ファイルシステムは、再び整合した状態になると新しいマスタとクライアントにより自動的にマウントされ、アプリケーションの処理が再開されます。

Cluster File System クライアントノードはファイルシステムメタデータを直接更新しないので、Cluster File System クライアントノードに障害が起きても、メタデータの修復は必要ありません。したがって、クライアントノードの障害からの Cluster File System のリカバリは、マスタノードの障害からのリカバリに比べ、高速になります。



SANPoint Foundation Suite ファイルシステムを含む
Cluster Volume Manager ポリユーム

図 6 : VERITAS Cluster File System に統合されているコンポーネント

Cluster File System は、Cluster File System がメンバーシップの判別とノード間通信のために、GAB (Group Membership and Atomic Broadcast) プロトコルと LLT (Low Latency Transport) プロトコルという 2 つのプロトコルを使用します。GAB と LLT は VERITAS Cluster Server 固有のプロトコルで、ファイバチャネルトランスポート (オーバヘッドと待ち時間を最小限に抑えるため) またはイーサネットデータリンクプロトコル上に直接導入できます。

GAB は、クラスタ全体にブロードキャストされたメッセージが正しく受信され (つまり確認され)、すべてのノードにより同じ順序で受信されることを保証するという意味で、一つのブロードキャストプロトコルです。GAB の主な用途はメンバーシップサービスを提供することで、クラスタ全体に提供するだけでなく、Cluster File System インスタンスなど、メンバーシップサービスを実行するアプリケーショングループにも提供します。GAB のメンバーシップサービスを使用すると、起動とシャットダウンを正しい順序で行うことができます。

LLT は、その名が示すように、多数の小さなメッセージ (分散ロックトラフィックの特徴) を単純なネットワークポロジ (クラスタの一般的な特徴) で高速配信するために最適化されています。

GAB と LLT は、どちらもクラスタ内のすべてのサーバを接続した冗長データリンクで動作するように設計されています。Cluster Server は、一つの通信リンクの障害によりクラスタの分割が起きる可能性を最小限に抑えるために、冗長クラスタ通信リンクを必要とします。

ベリタスソフトウェア株式会社

〒100-0011 東京都千代田区千代田 2-2-1 丸の内ビルディング 22F
www.veritas.com/jp